Theoretical Computer Science 31 (1984) 317–335
North-Holland

SOME RESULTS ON SUBCLASS CONTAINMENT PROBLEMS FOR SPECIAL CLASSES OF DPDA'S RELATED TO NONSINGULAR MACHINES

Michio OYAMAGUCHI

Department of Electrical Engineering, Department of Electronics, Faculty of Engineering, Mie University, 1515 Kamihama Tsu 514, Japan

Communicated by R. Book
Received April 1983
Revised December 1983

Abstract. A context-free language is said to be weakly (w-)nonsingular if it is accepted by a nonsingular deterministic pushdown automaton in the sense of Oyamaguchi, Inagaki and Honda (1980). It is undecidable whether a deterministic pushdown automaton (dpda) accepts a w-nonsingular language and whether a dpda is nonsingular in the sense of Valiant (1973). Next, the class of super-nonsingular dpda's (which is a subclass of w-nonsingular dpda's) is introduced. It is decidable whether a dpda is super-nonsingular and whether a dpda accepts a super-nonsingular language. As a consequence, the problem of deciding whether a dpda accepts an LL(k) language reduces to the problem of deciding whether a super-nonsingular dpda accepts an LL(k) language.

1. Introduction

Both the equivalence problem and the subclass containment problems for dpda's have received much attention in recent years. The equivalence problem has been shown to be decidable for several subclasses of dpda's [4, 6, 8–10, 13, 14, 16–19], although it remains open for general dpda's.

On the other hand, many of the subclass containment problems remain open. However, several contributions have been made to the problems [1, 3, 5, 11, 15, 20]. The containment problem relative to a class $\mathscr{C}$, written as containment (dpda, $\mathscr{C}$), is the problem of deciding for a dpda M whether there exists a machine in class $\mathscr{C}$ accepting the same language as M. For many subclasses $\mathscr{C}$ of dpda's, to decide containment (dpda, $\mathscr{C}$) is known to be at least as difficult as to decide the equivalence problem for $\mathscr{C}$ [3].

The decidability of containment (dpda, the class of finite automata) was first proven by Stearns [15]. Improvements of the algorithm were made by Valiant [20] and Courcelle [1]. Oyamaguchi et al. [11, 12] established the decidability of containment (dpda, the class of real-time strict dpda's) and containment (dpda, the class of simple dpda's).

In this paper we consider two subclass containment problems for dpda's, that is, containments (dpda, WN_0) and (dpda, SN_0) where WN_0 is the class of weakly (w-)nonsingular dpda's defined by Oyamaguchi et al. [12], and SN_0 is the class of super-nonsingular dpda's introduced in this paper. We show that containment (dpda, SN_0) is decidable, but containment (dpda, WN_0) is undecidable. These classes are properly included in the class R_0 of real-time strict dpda's and properly include the class LL(k) of LL(k) acceptors [14].

For this purpose we first prove that it is undecidable whether a dpda in R_0 is w-nonsingular (Theorem 3.11). Next, we show that containment (dpda in R_0, WN_0) reduces to the problem of deciding whether a dpda in R_0 is w-nonsingular (Lemma 4.5). We also show that containment (dpda in R_0, SN_0) reduces to the problem of deciding whether a dpda in R_0 is super-nonsingular (Lemma 5.4) and it is decidable whether a dpda in R_0 is super-nonsingular (Theorem 5.11). Thus, we obtain the main results of this paper. As a consequence, containment (dpda, LL(k)), which remains open, reduces to containment (dpda in SN_0, LL(k)).

Further, we show that the undecidability result of w-nonsingularity implies that it is undecidable whether a dpda is a nonsingular machine introduced by Valiant [18] (our Theorem 3.12). This is a negative solution to the problem posed in [2]. Some comments concerning containment (dpda, the class of nonsingular machines), which remains open, are made at the end of Section 4.

2. Definitions and notations

We use ε to denote the empty string and $\emptyset$ to denote the empty set. For a set X we let $|X|$ denote the cardinality of X, and for a string α we let $|\alpha|$ denote the length of α. We use $[k]$ to denote $\{1, 2, \ldots, k\}$ and $\mathbb{N}$ to denote the set of nonnegative integers.

A dpda is a sextuple $M = (Q, \Gamma, \Sigma, \Delta, c_s, F)$ where (1) Q, Γ and Σ are respectively the finite sets of states, stack symbols and input symbols, (2) c_s, the *initial mode*, is in $Q \times \Gamma$, (3) Δ, the set of *transition rules*, is a finite subset of $Q \times \Gamma \times (\Sigma \cup \{\varepsilon\}) \times Q \times \Gamma^*$, and (4) F, the set of accepting modes, is a subset of $Q \times (\Gamma \cup \{\varepsilon\})$. A transition (q, A, π, q', v), written $(q, A) \xrightarrow{\pi} (q', v)$, has a *mode* (q, A) and an *input* π. The set Δ satisfies the following conditions. For (q, A) in $Q \times \Gamma$, if there is no transition with mode (q, A) and input ε, then for each a in Σ there is at most one transition with mode (q, A) and input a, in which case (q, A) is called a *reading mode*. Otherwise, there is a unique transition with mode (q, A) and input ε, in which case (q, A) is called an ε *mode*.

A *configuration* of M is a member $c = (q, u)$ of $Q \times \Gamma^*$, and the *height* of c is $|c| = |u|$. If $u = \varepsilon$, the mode of c is (q, u) and otherwise the mode of c is (q, A), where A is the rightmost (i.e., top) symbol of u. If $(q, A) \xrightarrow{\pi} (q', v)$ in Δ, we write the *computation* from (q, uA) as $(q, uA) \xrightarrow{\pi} (q', uv)$. A sequence of computations $c_0 \xrightarrow{\pi_1} c_1 \to \cdots \xrightarrow{\pi_n} c_n$ is written as $c_0 \xrightarrow{\alpha} c_n$, where $\alpha = \pi_1 \ldots \pi_n$.

An input string α is *accepted from configuration* c if and only if there exists a computation $c \to^{\alpha} c'$ for some c' with an accepting mode. The set of words accepted from c is denoted by $L(c)$. Two configurations c_1 and c_2 are *equivalent*, denoted $c_1 \equiv c_2$, if $L(c_1) = L(c_2)$. The language accepted by M is $L(M) = L(c_s)$.

A configuration c is *reachable* from a configuration c' if $c' \to^{\alpha} c$ for some string α, and reachable if it is reachable from the initial configuration c_s. A configuration c is *live* if $L(c) \neq \emptyset$. For a live configuration c, let $\min(c) = \mathrm{Min}\{|\alpha| \mid \alpha \text{ in } L(c)\}$, the length of a shortest string accepted from c.

A computation $c \to^{\alpha} c'$ is written as $c \uparrow_{-\rho}(\alpha)\, c'$ for $\rho \geq 0$ if throughout the computation the stack height is at least $|c| - \rho$. If $\rho = 0$, it is written as $c \uparrow(\alpha)\, c'$. A computation $c \to^{\alpha} c'$ is written as $c \downarrow(\alpha)\, c'$ if throughout the computation the stack height is at least $|c'|$ and c' is the only configuration of height $|c'|$.

Let k_0 be the smallest number with the following property: For q, q' in Q, A in Γ, v in Γ^* with $|v| \leq 2$, if (q', v) is reachable from (q, A), then $(q, A) \to^{\alpha} (q', v)$ for some input string α with $|\alpha| + 1 \leq k_0$.

An input string α is *minimal* for a configuration c if α satisfies the following condition: For A in Γ, q, q' in Q and an initial segment $\alpha_1\alpha_2$ of α, if $c \to^{\alpha_1} (q, wA) \downarrow(\alpha_2)\, (q', w)$, then $|\alpha_2| < k_0$.

Let $c = (p, w)$ and $c' = (p', ww')$ be two configurations. If there exists a reachable configuration $d = (q, wA)$, A in Γ such that $d \downarrow(\beta)\, c$ and $d \uparrow(\beta')\, c'$ for some strings β, β', then the pair $\{c, c'\}$ is *strongly reachable via* d or strongly reachable.

The *size* of M is $|M| = |Q| \cdot |\Gamma| \cdot |\Sigma|^{*} \cdot m$ where m is the maximum length of stack strings appearing in Δ.

A *real-time* dpda is a dpda with no ε modes, and a *real-time strict* dpda is a real-time dpda with empty stack acceptance. Let R_0 be the class of real-time strict dpda's. Let LL(k) and S_0 be the classes of LL(k) acceptors [14] and simple dpda's [8], respectively. Let D_0 be the class of dpda's with empty stack acceptance.

A dpda M in D_0 is *nonsingular* [2] if there exists a positive constant n_0 with the following property: For any two reachable configurations $c = (p, w)$ and $c' = (p', ww')$ where $|w'| > n_0$, either $L(c) \neq L(c')$ or $L(c) = L(c') = \emptyset$. Let N_0 be the class of nonsingular dpda's. It is known that family relationships $\mathcal{L}(S_0) \subsetneq \mathcal{L}(\mathrm{LL}(k)) \subsetneq \mathcal{L}(N_0) \subsetneq \mathcal{L}(R_0) \subsetneq \mathcal{L}(D_0)$ [14, 18] where $\mathcal{L}(\mathcal{C}) = \{L(M) \mid M \text{ in } \mathcal{C}\}$ for each subclass $\mathcal{C}$ of dpda's.

3. Undecidability of nonsingularity and w-nonsingularity

We now define the class of weakly (w-)nonsingular dpda's which includes the class of nonsingular dpda's [2]. Then we show that it is undecidable whether a dpda is a w-nonsingular machine. As a corollary we show that it is undecidable whether a dpda is nonsingular. This is a negative solution to the problem posed in [2].

Definition 3.1. A dpda M in D_0 is weakly (w-)nonsingular if there exists a positive constant n_0 with the following property:

(A) For any pair of strongly reachable configurations $c = (p, w)$ and $c' = (p', ww')$, if $c \equiv c'$ and $L(c) \neq \emptyset$, then $|w'| \leq n_0$.

Henceforth, a number n_0 with property (A) is said to be a w-nonsingularity constant of dpda M.

Definition 3.2. Let WN_0 be the class of w-nonsingular dpda's.

Definition 3.3. A language L is w-nonsingular if $L \in \mathscr{L}(\mathrm{WN}_0)$.

Note. The only distinction between the definitions of w-nonsingularity and nonsingularity is that the term 'strongly' is present or not. The distinction will lead us to the following: class WN_0 is R_0-closed (see Definition 4.1), but class N_0 is not R_0-closed.

Lemma 3.4. $\mathscr{L}(N_0) \subseteq \mathscr{L}(\mathrm{WN}_0) \subsetneq \mathscr{L}(R_0)$.

Proof. It is clear that $\mathscr{L}(N_0) \subseteq \mathscr{L}(\mathrm{WN}_0)$ by the definitions of N_0 and WN_0. $\mathscr{L}(\mathrm{WN}_0) \subseteq \mathscr{L}(R_0)$ is also obvious, since w-nonsingular dpda's are quasi-real-time (i.e., the length of any sequence of consecutive ε-moves is bounded). To show proper inclusion $\mathscr{L}(\mathrm{WN}_0) \subsetneq \mathscr{L}(\mathrm{R}_0)$, consider $L = \{a^n bc^n \mid n \geq 1\} \cup \{a^n dc^{2n} \mid n \geq 1\}$. It is obvious that $L \in \mathscr{L}(R_0)$, but we can show that $L \notin \mathscr{L}(\mathrm{WN}_0)$ whose proof is the same as that of $L \notin \mathscr{L}(N_0)$ [18]. □

We will show that the problem of deciding whether a dpda is w-nonsingular reduces to the halting problem of Turing machines (TM). Thus, the former is undecidable. To show this we first associate a language to an instance of Post's correspondence problem (PCP) which is closely related to computations of a TM [7, p. 196].

Let $P = (U = \{u_1, \ldots, u_k\}, V = \{v_1, \ldots, v_k\})$ be an instance of PCP where $u_i, v_i \in \{a, b\}^*$, $1 \leq i \leq k$. We define the language $L_P = L_U \cup L_V$ where, for $X \in \{U, V\}$,

$$L_X = \{i_1 \ldots i_n f_X x_{i_n}^{\mathrm{R}} \ldots x_{i_1}^{\mathrm{R}} \mid i_1 = 1, i_j \in [k], x_{i_j} \in X, 1 \leq j \leq n\}.$$

Here, f_U, f_V are new symbols and x^{R} is the reverse of x. Note that we require $i_1 = 1$ if $i_1 \ldots i_n f_X w \in L_X$, that is, we consider a modified version of PCP (MPCP) which is known to be undecidable [7, p. 196]. A dpda M_P accepting L_P is constructed as follows.

Construction 1. $M_P = (Q, \Gamma, \Sigma, \Delta, c_s, F)$:

$$Q = \{q_0\} \cup Q_U \cup Q_V \quad \text{where}$$

$$Q_U = \{q_\alpha \mid u_i^R = \alpha\beta \text{ for some } \beta \text{ in } \{a,b\}^* \text{ and } u_i \in U\},$$

$$Q_V = \{p_\alpha \mid v_i^R = \alpha\beta \text{ for some } \beta \text{ in } \{a,b\}^* \text{ and } v_i \in V\},$$

$\Gamma = \{Z_0\} \cup [k]$,

$\Sigma = \{a, b, f, g\} \cup [k]$ where $f = f_U$, $g = f_V$,

$c_s = (q_0, Z_0)$,

$F = \{(q_\varepsilon, \varepsilon), (p_\varepsilon, \varepsilon)\}$,

Δ: For each $x \in \{a, b\}$, $i, j \in [k]$,

$(q_0, Z_0) \to^1 (q_0, 1)$,

$(q_0, i) \to^j (q_0, ij)$,

$(q_0, i) \to^f (q_\varepsilon, i)$,

$(q_0, i) \to^g (p_\varepsilon, i)$,

$(q_\alpha, i) \to^x (q_\varepsilon, \varepsilon)$ if $\alpha x = u_i^R$,

$(q_\alpha, i) \to^x (q_{\alpha x}, i)$ if $|\alpha x| < |u_i|$ and αx is a prefix of u_i^R,

$(p_\alpha, i) \to^x (p_\varepsilon, \varepsilon)$ if $\alpha x = v_i^R$.

$(p_\alpha, i) \to^x (p_{\alpha x}, i)$ if $|\alpha x| < |v_i|$ and αx is a prefix of v_i^R.

Typical computations of machine M_P are, for example,

$$(q_0, Z_0) \to^{1 i_1 \ldots i_n} (q_0, 1 i_1 \ldots i_n) \quad \text{where } i_j \in [k] \text{ for } 1 \le j \le n,$$

$$(q_0, 1 i_1 \ldots i_n) \to^f (q_\varepsilon, 1 i_1 \ldots i_n) \to^{u_{i_n}^R} (q_\varepsilon, 1 i_1 \ldots i_{n-1}),$$

$$(q_0, 1 i_1 \ldots i_n) \to^g (p_\varepsilon, 1 i_1 \ldots i_n) \to^{v_{i_n}^R} (p_\varepsilon, 1 i_1 \ldots i_{n-1}).$$

Note that $M_P \in R_0$, and any reachable configuration c of M_P is in $Q \times [k]^*$ or $c = (q_0, Z_0)$. Throughout this section we deal with a fixed MPCP $P = (U = \{u_1, \ldots, u_k\}, V = \{v_1, \ldots, v_k\})$ and dpda M_P associated to P.

The next lemma is a technical one used for the proof of Lemmas 3.7 and 3.9.

Lemma 3.5. *Let $c = (r, w)$ and $c' = (r', ww')$ be reachable configurations of M_P such that $c \equiv c'$, $|w'| > 0$ and $L(c) \neq \emptyset$. Then, either $c \in Q_U \times [k]^*$ and $c' \in Q_V \times [k]^*$, or $c \in Q_V \times [k]^*$ and $c' \in Q_U \times [k]^*$.*

Proof. It is obvious that $c \neq (q_0, Z_0)$ since $(q_0, Z_0 w')$, $|w'| > 0$, is not reachable. So let $c = (r, i_1 \ldots i_m)$ and let $c' = (r', i_1 \ldots i_m w')$ where $r, r' \in Q$, $i_j \in [k]$, $1 \le j \le m$, and $w' \in [k]^+$. Then, $r \neq q_0$ holds. For the sake of contradiction, suppose $r = q_0$. If $r' = q_0$, then $\min(c) < \min(c')$ by $|w'| > 0$, so that $c \not\equiv c'$, a contradiction. If $r' \in Q_U \cup Q_V$, then, obviously, $c \not\equiv c'$, because c only accepts strings containing f or g, but c' does not. In either case we have a contradiction. So let $r \in Q_U \cup Q_V$. By the above arguments, $r \neq q_0$ implies $r' \neq q_0$.

Let $r \in Q_U$. Then, we show $r' \in Q_V$. To the contrary, suppose that $r' \in Q_U$. Then, by the definition of M_P, c, c' accept exactly one input string, and there exists a computation $(r', i_m w') \rightarrow^{\alpha} (r, i_m)$. By $|w'| > 0$, $|\alpha| > 0$ holds, so that $\min(c) < \min(c')$, a contradiction. Thus, $r' \in Q_V$. Similarly, $r \in Q_V$ implies $r' \in Q_U$. □

Definition 3.6. Let $l_0 = \mathrm{Max}(|u_1|, \ldots, |u_k|, |v_1|, \ldots, |v_k|)$ where $u_j \in U$, $v_j \in V$ for $1 \leq j \leq k$ and $P = (U, V)$.

Lemma 3.7. *M_P is* w*-nonsingular if and only if M_P is nonsingular.*

Proof. '*If*' *part.* Obvious.

'*Only if*' *part.* For reachable configurations $c = (r, w)$ and $c' = (r', ww')$, assume that $c \equiv c'$, $|w'| > 0$ and $L(c) \neq \emptyset$. Then, $w, w' \in [k]^*$ and either $r \in Q_U$ and $r' \in Q_V$ or $r \in Q_V$ and $r' \in Q_U$ by Lemma 3.5. Without loss of generality we assume $r \in Q_U$ and $r' \in Q_V$. So let $r = q_\delta$ and $r' = p_{\delta'}$ for some δ, δ' in $\{a, b\}^*$. Let $e = (q_0, w)$. Obviously, e is reachable, since $|w| > 0$ by $c \equiv c'$ and $|w'| > 0$. By the definition of M_P, we have the following computations:

$$e = (q_0, w) \rightarrow^{f\delta} c = (q_\delta, w),$$

$$e = (q_0, w) \uparrow (w'g\delta') \, c' = (p_{\delta'}, ww').$$

For some string γ of length $\leq l_0$, $c \downarrow (\gamma) \, d$ and $|c| - |d| = 1$. Let $c' \rightarrow^{\gamma} d'$. Then, $d \equiv d'$ holds by $c \equiv c'$. If $|w'| > l_0$, then $e \uparrow (w'g\delta'\gamma) \, d'$ holds. Hence, the pair $\{d, d'\}$ is strongly reachable via e. Since $||d| - |d'|| \leq n_0$ where n_0 is a w-nonsingularity constant, it follows that $|w'| = |c'| - |c|$ is bounded by a fixed constant. Thus, M_P is nonsingular. □

Definition 3.8 ([7]). A pair (α, β) of strings α, β in $\{a, b\}^*$ is a partial solution to MPCP P if there exist $i_1, \ldots, i_n$ in $[k]$ such that $\alpha = u_1 u_{i_1} \ldots u_{i_n}$ is a prefix of $\beta = v_1 v_{i_1} \ldots v_{i_n}$ or β is a prefix of α.

Lemma 3.9. *Machine M_P is* w*-nonsingular if and only if there exists a positive constant m_0 with the following property:*

(B) *For any two strings α, β in $\{a, b\}^*$, if (α, β) is a partial solution to P, then $||\alpha| - |\beta|| \leq m_0$.*

Proof. '*Only if*' *part.* Without loss of generality, we assume that $\beta = \alpha\gamma$ for some $\gamma \in \{a, b\}^*$ where (α, β) is a partial solution to P. Let $\xi = 1 i_1 \ldots i_n$ where $\alpha = u_1 u_{i_1} \ldots u_{i_n}$ and $\beta = v_1 v_{i_1} \ldots v_{i_n}$. Then, $\xi f \alpha^R$ and $\xi g \beta^R$ belong to $L(M_P)$. Consider the following computations:

$$c_s \to^{\xi} (q_0, \xi) \to^{f} c_2 = (q_\varepsilon, \xi_1\xi_2),$$

$$c_s \to^{\xi} (q_0, \xi) \to^{g\gamma^R} c_1 = (p_\delta, \xi_1),$$

where $\xi = \xi_1\xi_2$ and $\delta \in \{a, b\}^*$. Then, $L(c_1) = L(c_2) = \{\alpha^R\}$, since $\xi f\alpha^R$, $\xi g\beta^R \in L(M_P)$ and $\beta = \alpha\gamma$. Also, the following holds:

$$|c_2| - |c_1| = |\xi_2| \geq |\gamma|/l_0 - 1 \tag{$\star$}$$

because any string of γ^R of length l_0 pops at least one stack symbol. Since M_P is w-nonsingular, M_P is nonsingular by Lemma 3.7. So, $c_2 = (q_\varepsilon, \xi_1\xi_2) \equiv c_1 = (p_\delta, \xi_1)$ and $L(c_1) \neq \emptyset$ imply that $|\xi_2|$ is bounded by a fixed constant. Hence, inequality ($\star$) ensures that $|\gamma| = |\beta| - |\alpha|$ is bounded by a fixed constant.

'*If' part.* Let $\{c = (r, w), c' = (r', ww')\}$ be a pair of strongly reachable configurations such that $c \equiv c'$ and $L(c) \neq \emptyset$. Then, $w, w' \in [k]^*$ and either $r \in Q_U$ and $r' \in Q_V$ or $r \in Q_V$ and $r' \in Q_U$ by Lemma 3.5. Without loss of generality we assume $r \in Q_U$ and $r' \in Q_V$. So let $r = q_\delta$ and $r' = p_\xi$ for some δ, ξ in $\{a, b\}^*$. Then, $q_\delta = q_\varepsilon$ holds because $\{c = (q_\delta, w), c' = (p_\xi, ww')\}$ is strongly reachable, that is, $e \downarrow(\eta)\, c$ and $e \uparrow(\eta')\, c'$ for some reachable configuration e and input strings η, η', so that by the definition of M_P, $e \downarrow(\eta)\, c = (q_\delta, w)$ implies $q_\delta = q_\varepsilon$. Thus, c and c' can be written as

$$c = (q_\varepsilon, 1 i_1 \ldots i_m), \qquad c' = (p_\xi, 1 i_1 \ldots i_m w'),$$

where $\xi \in \{a, b\}^*$, $w' \in [k]^+$ and $i_j \in [k]$, $1 \leq j \leq m$. Since c and c' accept exactly one input string, let $L(c) = L(c') = \{\alpha\}$. Then, $L(c) = \{\alpha\}$ implies $\alpha = u_{i_m}^R \ldots u_{i_1}^R u_1^R$. Consider the accepting computation from c':

$$c' = (p_\xi, 1 i_1 \ldots i_m w') \downarrow(\gamma) (p_\varepsilon, 1 i_1 \ldots i_m) \downarrow(\beta) (p_\varepsilon, \varepsilon). \tag{$\star\star$}$$

Then, $\gamma\beta = \alpha$ by $L(c) = L(c')$, and $\beta = v_{i_m}^R \ldots v_{i_1}^R v_1^R$. So, $\beta^R = v_1 v_{i_1} \ldots v_{i_m}$ is a prefix of $\alpha^R = u_1 u_{i_1} \ldots u_{i_m}$, so that property (B) ensures $|\gamma| = |\alpha| - |\beta| \leq m_0$. Note that since M_P makes the computation ($\star\star$) in real-time, $|w'| \leq |\gamma|$. Hence, $|w'| \leq |\gamma| \leq m_0$ holds. Thus, m_0 is a w-nonsingularity constant of M_P, because $|w'| \leq m_0$ for any pair $\{c = (r, w), c' = (r', ww')\}$ of strongly reachable configurations such that $c \equiv c'$ and $L(c) \neq \emptyset$. □

By Lemmas 3.7 and 3.9, the problems of deciding whether a dpda in R_0 is w-nonsingular and whether a dpda in R_0 is nonsingular reduce to the problem of deciding for an MPCP P whether there exists a constant m_0 with property (B) of Lemma 3.9. So our goal is to show the undecidability of the latter problem.

Let M be a TM and let w_0 be a given input. Then, an instance P of MPCP is constructed from M and w_0 by the same way as that of [7, p. 197], where lists of Groups III, IV in [7, p. 197] are unnecessary in our arguments. It is known that if there is a valid sequence of m instantaneous descriptions (ID's) $I_1 \vdash_M I_2 \vdash_M \cdots \vdash_M I_m$, $m > 1$, then there exists a partial solution (α, β) such that

$\alpha = \# I_1 \# \cdots \# I_{m-1} \#$ and $\beta = \# I_1 \# \cdots \# I_{m-1} \# I_m \#$. Conversely, if (α', β') is a partial solution, then α', β' are prefixes of α, β, respectively. Moreover, if $|\beta'| = |\beta|$, then $(\alpha', \beta') = (\alpha, \beta)$ holds [7]. We are now ready to prove the following lemma.

Lemma 3.10. *It is undecidable whether there exists a positive constant m_0 with property* (B) *of Lemma* 3.9.

Proof. We assume, to the contrary, that the problem is decidable. Then, we show that it would be decidable whether a TM M accepts a given word w_0. Thus, we have a contradiction.

We first note that if there exists a positive constant m_0 with property (B), then, for each ID I_n, $|I_n| \leq m_0$ holds, because there exists a partial solution (α, β) such that $\beta = \alpha I_n \#$, by the previous arguments. So, TM M halts on input w_0, or else M enters a loop. Since $|I_n| \leq m_0$, we can eventually check whether either of the two possibilities occurs. Thus, we can decide whether M accepts w_0, since M accepts w_0 iff M halts in w_0 and enters an accepting state. On the other hand, if there is no positive constant m_0 with property (B), then for any positive constant m_0 there exists a partial solution (α', β') such that $||\alpha'| - |\beta'|| > m_0$. Since α', β' are prefixes of α, β such that $(\alpha = \# I_1 \# \cdots \# I_{n-1} \#, \beta = \# I_1 \# \cdots \# I_{n-1} \# I_n \#)$ is a partial solution for some $n > 1$, we choose such a least n. Let $\alpha'' = \# I_1 \# \cdots \# I_{n-2} \#$ and let $\beta'' = \# I_1 \# \cdots \# I_{n-2} \# I_{n-1} \#$. Then, (α'', β'') is the only partial solution whose larger string is as long as $|\beta''|$. Further, $\alpha' = \alpha'' \gamma$ and $\beta' = \beta'' \gamma'$ holds for some strings γ, γ'. Since γ contains at most one state symbol, by the definition of Group II in [7, p. 197], $|I_{n-1}| \geq m_0$ holds. Thus, sizes of ID's are unbounded. Hence, TM M diverges, that is, M consumes an infinite symbol of the tape. Thus, M does not accept w_0.

In either case we can decide whether M accepts w_0. Thus, it would be decidable whether a TM accepts a given word, a contradiction. □

By the above lemmas we have the following main results of this section.

Theorem 3.11. *It is undecidable, for a given dpda M in R_0, whether M is w-nonsingular.*

The proof is obvious by Lemmas 3.9 and 3.10, since dpda M_P in Lemma 3.9 is real-time strict.

Thus, it is undecidable whether a dpda is w-nonsingular.

By Lemmas 3.7, 3.9 and 3.10 we also have the following theorem.

Theorem 3.12. *It is undecidable, for a given dpda M in R_0, whether M is nonsingular.*

Note. The class of nonsingular dpda's was introduced as an example of a class of machines for which Valiant's alternate stacking technique succeeds [18]. Further arguments concerning this technique are made in [9, 21, 22].

4. Undecidability of containment (dpda, WN_0)

In this section we show that it is undecidable whether a dpda in R_0 accepts a w-nonsingular language. To show this we prove that the class WN_0 is an R_0-closed class which is defined as follows.

Definition 4.1. A subclass $\mathscr{C}$ of dpda's is closed under a subclass $\mathscr{C}'$ of dpda's if, for any dpda M in $\mathscr{C}'$, $L(M) \in \mathscr{L}(\mathscr{C})$ implies $M \in \mathscr{C}$, and R_0-closed if, for any dpda M in R_0, $L(M) \in \mathscr{L}(\mathscr{C})$ implies $M \in \mathscr{C}$.

By the above definition, if the class WN_0 is R_0-closed, then for a dpda M in R_0, $L(M) \in \mathscr{L}(WN_0)$ iff $M \in WN_0$. So containment (dpda in R_0, WN_0) reduces to the problem of deciding whether a dpda in R_0, is w-nonsingular. Thus, by Theorem 3.11 of the previous section, containment (dpda in R_0, WN_0) is undecidable.

To show that WN_0 is R_0-closed, we need a definition and some lemmas. Henceforth we are dealing with a fixed real-time strict dpda $M = (Q, \Gamma, \Sigma, \Delta, c_s, F)$ in R_0. Throughout this section, if $L(M)$ is w-nonsingular, then a dpda $M' = (Q', \Gamma', \Sigma, \Delta', d_s, F')$ in R_0 stands for a w-nonsingular machine such that $L(M) = L(M')$.

Definition 4.2 ([9]). Let $M = (Q, \Gamma, \Sigma, \Delta, c_s, F)$ in R_0. Then, $k_{0,M}$ is the smallest number with the following property:

> For any q, q' in Q, A in Γ, v in Γ^* with $|v| \leq 2$, if (q', v) is reachable from (q, A), then $(q, A) \to^{\alpha} (q', v)$ for some input string α with $|\alpha| + 1 \leq k_0$.

The following lemma was proved in [9].

Lemma 4.3 ([9]). *Let M, M' in R_0 be equivalent. Then, there exists a positive constant m such that if c_1 and c_2 are equivalent, reachable and live configurations of M and M', and $c_1 \uparrow(\alpha)\, c_1'$ for a string α, then $c_2 \uparrow_{-m}(\alpha)\, c_2'$. Moreover, such an m can be calculated and depends only on the sizes of M and M'.*

Without loss of generality, we assume that all reachable configurations are live [9], hereafter.

The next lemma says the following: Let dpda's M and M' in R_0 be equivalent and let a pair $\{c_1, c_2\}$ be strongly reachable configurations of M via a configuration c. Then, if $c_s \to^{\alpha} c$, $c \downarrow(\beta)\, c_1$ and $c \uparrow(\gamma)\, c_2$, and the difference between $|c_2|$ and $|c_1|$ is sufficiently large, then for the corresponding computations $d_s \to^{\alpha} d$, $d \to^{\beta} d_1$ and $d \to^{\gamma} d_2$ of M', the difference between $|d_2|$ and $|d_1|$ is larger than a logarithm of the difference between $|c_2|$ and $|c_1|$ (multiplied by a fixed constant) and $d \uparrow_{-m}(\gamma)\, d_2$ holds.

Lemma 4.4. *Let M, M' in R_0 be equivalent. Let a pair $\{c_1, c_2\}$ of configurations of M be strongly reachable via a configuration c. Let $|c_2| - |c_1| > n$. Here, n is larger than*

$2^{p(m,k_0)}$ *where* $p(m, k_0)$ *is a polynomial of* m *and* k_0, m *is the constant of Lemma* 4.3 *and* k_0 *is* $\mathrm{Max}(k_{0,M}, k_{0,M'})$ *where* $k_{0,M}$ *and* $k_{0,M'}$ *are constants of Definition* 4.2. *Let* l *be some positive constant depending only on the sizes of* M *and* M'. *Then, there exists a pair* $\{d_1, d_2\}$ *of configurations of* M' *such that*

(i) $d_1 \equiv c_1$ *and* $d_2 \equiv c_2$,

(ii) $|d_2| - |d_1| \geq l \log n$,

(iii) *for some configuration* d *and inputs* β, γ, *where* $|\beta| \leq k_0$, *it holds that* $c \uparrow(\gamma)\, c_2$, $d \uparrow_{-m}(\gamma)\, d_2$ *and* $c \downarrow(\beta)\, c_1$, $d \rightarrow^{\beta} d_1$.

Proof. Let c_s and d_s be the initial configurations of M and M', respectively. Consider the following computations,

$$C_0 : c_s \rightarrow^{\alpha} c, \qquad C_1 : c \downarrow(\beta)\, c_1, \qquad C_2 : c \uparrow(\gamma)\, c_2,$$

where β and γ are respectively shortest imputs such that $c \downarrow(\beta)\, c_1$ and $c \uparrow(\gamma)\, c_2$. Note that since $\{c_1, c_2\}$ is strongly reachable via c, $|c| - |c_1| = 1$ and, by Definition 4.2, $|\beta| \leq k_0$. Consider the corresponding computations of M' to C_0, C_1 and C_2,

$$D_0 : d_s \rightarrow^{\alpha} d, \qquad D_1 : d \rightarrow^{\beta} d_1, \qquad D_2 : d \rightarrow^{\gamma} d_2.$$

Then, we show that the pair $\{d_1, d_2\}$ satisfies (i), (ii), (iii) of this lemma. By $c_s \equiv d_s$, we have $c \equiv d$, $c_1 \equiv d_1$ and $c_2 \equiv d_2$. Thus, (i) holds. Further, (iii) holds, because $d \uparrow_{-m}(\gamma)\, d_2$ by Lemma 4.3 and $|\beta| \leq k_0$ (see Fig. 1).

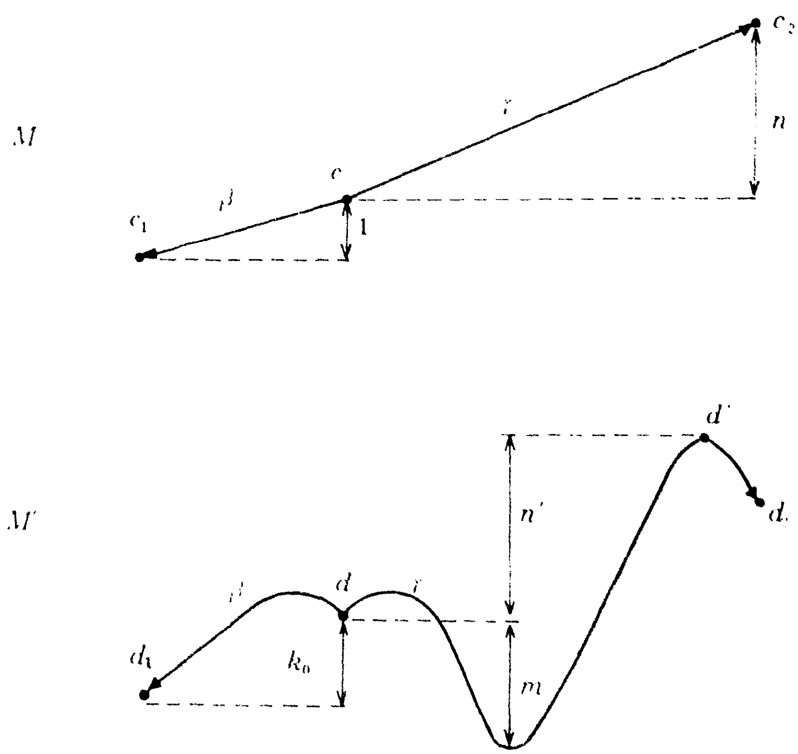

Fig. 1. Derivations in the proof of Lemma 4.4.

To show (ii) we first note that the number of pairwise inequivalent configurations appearing within computation $C_2: c \uparrow(\gamma)\, c_2$ is at least $(|c_2|-|c|)/k_0$, that is, n/k_0, because if $c' \uparrow(\delta)\, c''$ is a subcomputation of C_2 such that $|c''|-|c'| \geq k_0$, then $\min(c') < \min(c'')$. So, by $c \equiv d$, the same number (i.e., at least n/k_0) of pairwise inequivalent configurations must appear within computation $D_2: d \uparrow_{-m}(\gamma)\, d_2$. Hence, some configuration d' appearing within computation D_2 must have the height $|d'| \geq |d|+n'$ where $n/k_0 \leq |Q'|\,|\Gamma'|^{n'+m+1} \leq |M'|^{n'+m+2}$, because otherwise the number of different configurations appearing within D_2 would be less than n/k_0 by $d \uparrow_{-m}(\gamma)\, d_2$, a contradiction. Thus,

$$n'+m+2 \geq (\log|M'|)^{-1}(\log n - \log k_0).$$

That is, n' is larger than $l \log n$ for some fixed constant l depending only on the sizes of M, M'. Since the remaining computation from d' decreases the stack at most constant m by Lemma 4.3, and $\| d|-|d_1\| \leq k_0$ by $d \to^{\beta} d_1$ and $|\beta| \leq k_0$, it follows that $|d_2|-|d_1| \geq l \log n$ holds for some constant $l>0$. Thus, (ii) holds. □

Lemma 4.5. *If the language accepted by a machine M in R_0 is* w*-nonsingular, then M is a* w*-nonsingular dpda.*

Proof. Since $L(M)$ is w-nonsingular, there exists a w-nonsingular dpda $M'=(Q', \Gamma', \Sigma, \Delta', d_s, F')$ in R_0 such that $L(M)=L(M')$. Let n_0 be a w-nonsingularity constant of M'. We assume, to the contrary, that M is not a w-nonsingular dpda. That is, for any natural number n, there exists a strongly reachable pair of configurations $c_1=(p, w)$, $c_2=(p', ww')$ such that $c_1 \equiv c_2$ and $|w'|>n$. So, by Lemma 4.4, there exists a pair $\{d_1, d_2\}$ of configurations of M' satisfying (i), (ii), (iii) of Lemma 4.4. Note that $d_1 \equiv d_2$ holds by $c_1 \equiv c_2$.

Consider the following computations from equivalent configurations d_1 and d_2,

$$d_1 \downarrow(\delta)\, \bar{d}_1 \quad \text{and} \quad d_2 \to^{\delta} \bar{d}_2,$$

where δ is a shortest input such that $|d_1|-|\bar{d}_1| = m+k_0$. Here, k_0 is $\mathrm{Max}(k_{0,M}, k_{0,M'})$ and m is the constant of Lemma 4.3. Then, $|\delta| \leq k_0(m+k_0)$ holds, and, by Lemma 4.4(ii), $|d_2|-|d_1| \geq l \log n$, so that

$$|\bar{d}_2|-|\bar{d}_1| \geq l \log n - k_0(m+k_0) \;(=n_1).$$

We can choose n so that $n_1 > n_0$ holds. Then, obviously the pair $\{\bar{d}_1, \bar{d}_2\}$ is strongly reachable (see Fig. 2), $\bar{d}_1 \equiv \bar{d}_2$ and $|\bar{d}_2|-|\bar{d}_1| > n_0$. Hence, n_0 is not a w-nonsingularity constant of M', a contradiction. □

By Lemma 4.5, the class WN_0 is R_0-closed. Hence, for a dpda M in R_0, $L(M)$ is w-nonsingular iff M is w-nonsingular. By Theorem 3.11 we have the following main result of this section.

Theorem 4.6. *It is undecidable, for a dpda M in R_0, whether M accepts a* w*-nonsingular language.*

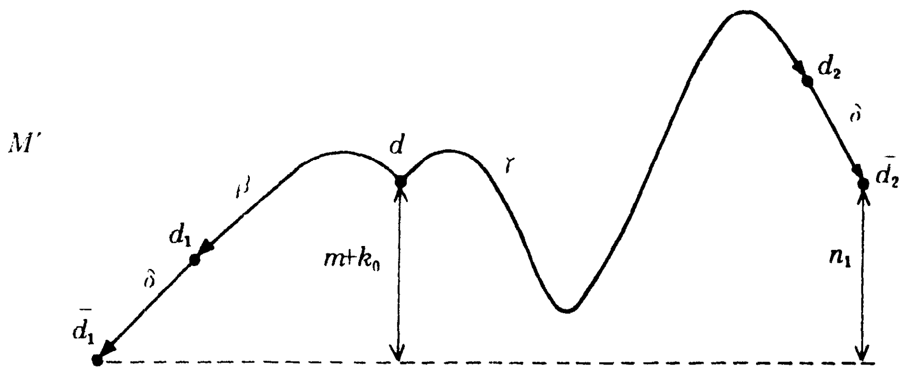

Fig. 2. Derivations in the proof of Lemma 4.5.

Thus, containment (dpda in R_0, WN_0) is undecidable. However, for a dpda M in R_0 and a given constant n, it is decidable whether n is a w-nonsingularity constant of M [12, Theorem 3.2].

For nonsingular dpda's we cannot have the result corresponding to Lemma 4.5, since we will now show that the class N_0 is not R_0-closed. To show this, consider a language $L_1 = \{a^n c^n \mid n \geq 1\} \cup \{d^n c^{2n} \mid n \geq 1\}$. Let $M = (Q, \Gamma, \Sigma, \Delta, c_s, F)$ be the dpda in R_0 such that for A, D in Γ, q_1, p_1, q_2, p_2 in Q, a, c, d in Σ and $n > 1$,

$$c_s \to^{a^n} (q_1, A^n) \to^{c} (p_1, A^{n-1}) \to^{c} (p_1, A^{n-2}),$$

$$c_s \to^{d^n} (q_2, D^n) \to^{cc} (p_2, D^{n-1}) \to^{cc} (p_2, D^{n-2}).$$

The precise definition is given as follows:

$$Q = \{q_0, q_1, q_2, p_1, p_2, p\}, \qquad \Gamma = \{z_0, A, D\}, \qquad \Sigma = \{a, c, d\},$$

$$c_s = (q_0, z_0), \qquad F = \{(p_1, \varepsilon), (p_2, \varepsilon)\}$$

and

$$\begin{aligned}\Delta = \{&(q_0, z_0) \to^{a} (q_1, A),\ (q_1, A) \to^{a} (q_1, AA),\\ &(q_1, A) \to^{c} (p_1, \varepsilon),\ (p_1, A) \to^{c} (p_1, \varepsilon),\\ &(q_0, z_0) \to^{d} (q_2, D), (q_2, D) \to^{d} (q_2, DD),\\ &(q_2, D) \to^{c} (p, D),\ (p, D) \to^{c} (p_2, \varepsilon),\\ &(p_2, D) \to^{c} (p, D)\}.\end{aligned}$$

It is obvious that $L(M) = L_1$ and M is nonsingular. Thus, L_1 is nonsingular. But, we can construct a dpda M' such that M' accepts L_1 and M' is not nonsingular. Let M' be the same as M except assuming $D = A$. Certainly, $L(M) = L_1$ and M' is not nonsingular, since $(p_2, A^n) \equiv (p_1, A^{2n})$ for any $n \geq 1$. Thus, N_0 is not R_0-closed. Hence, we have the following lemma.

Lemma 4.7. (i) *The class N_0 is not R_0-closed, but the class WN_0 is R_0-closed.*

(ii) $N_0 \subsetneq \mathrm{WN}_0$.

Note. We have only shown that the arguments in this section cannot be applied to containment (dpda, N_0). The problem remains open. Also, it remains open whether $\mathscr{L}(\mathrm{WN}_0) = \mathscr{L}(N_0)$. Note that $\mathscr{L}(\mathrm{WN}_0) \subseteq \mathscr{L}(N_0)$ implies the undecidability of containment (dpda, N_0) by Theorem 4.6, since $\mathscr{L}(N_0) \subseteq \mathscr{L}(\mathrm{WN}_0)$ is obvious.

5. Decidability of containment (dpda, SN_0)

We introduce a new subclass of dpda's, called super-nonsingular dpda's whose definition has a form similar to that of w-nonsingular dpda's.

Definition 5.1. A dpda M in D_0 is said to be super-nonsingular if there exists a positive constant n_0 with the following property:

(C) For any pair of strongly reachable configurations $c_1 = (p, w)$ and $c_2 = (p', ww')$, if c_1, c_2 are live and $\min(c_1) \geq \min(c_2)$, then $|w'| \leq n_0$.

Definition 5.2. Let SN_0 be the class of super-nonsingular dpda's.

Lemma 5.3. (i) $\mathscr{L}(\mathrm{SN}_0) \subsetneq \mathscr{L}(\mathrm{WN}_0)$.
(ii) $\mathscr{L}(\mathrm{LL}(k)) \subsetneq \mathscr{L}(\mathrm{SN}_0)$.

Proof. (i) It is obvious that $\mathscr{L}(\mathrm{SN}_0) \subseteq \mathscr{L}(\mathrm{WN}_0)$ by the above definition. To show that the inclusion is proper, consider $L_2 = \{a^n bc^n \mid n \geq 1\} \cup \{a^n de^{2n} \mid n \geq 1\}$. It is known that $L_2 \in \mathscr{L}(N_0)$ [18], so $L_2 \in \mathscr{L}(\mathrm{WN}_0)$. To show $L_2 \notin \mathscr{L}(\mathrm{SN}_0)$, let a dpda M in R_0 accept L_2. Consider computations of M, $c_s \to^{a^n} c$, $c \to^{b} c_1$ and $c \to^{de^n} c_2$ where c_s is the initial configuration of M and $n \geq 1$. Then, obviously $\min(c_1) = \min(c_2)$ and $|c_1| - |c_2|$ is not bounded by a fixed constant. So we can easily show that M is not super-nonsingular.

(ii) For an LL(k) acceptor M it is known that there exists a natural number l such that, for any pair of strongly reachable configurations $c_1 = (q, w)$ and $c_2 = (q', ww')$, $|\min(c_2) - \min(c_1) - \min(q', w')| \leq l$ [18, 14]. So if $\min(c_1) \geq \min(c_2)$, then $\min(q', w') \leq l$. Thus, l is a super-nonsingularity constant of M. To show that $\mathscr{L}(\mathrm{LL}(k)) \subsetneq \mathscr{L}(\mathrm{SN}_0)$, consider $L = \{a^n b^n \mid n \geq 1\} \cup \{a^n c^n \mid n \geq 1\}$. It is obvious that $L \in \mathscr{L}(\mathrm{SN}_0)$, but $L \notin \mathscr{L}(\mathrm{LL}(k))$ is known from [14]. $\square$

In this section we show that SN_0 is R_0-closed and it is decidable for a dpda M in R_0 whether M is super-nonsingular. Thus, containment (dpda in R_0, SN_0) is decidable. Since containment (dpda, R_0) is decidable [11] and $\mathscr{L}(\mathrm{SN}_0) \subseteq \mathscr{L}(R_0)$, it follows that it is decidable for a dpda M whether $L(M)$ is super-nonsingular. The results are compared with the undecidability ones concerning w-nonsingular dpda's in the previous section. Some comments will be made later. We first prove that the class SN_0 is R_0-closed. The proof is similar to that of Lemma 4.5 of the previous section.

Lemma 5.4. *For a dpda M in R_0, if $L(M)$ is super-nonsingular, then M is a super-nonsingular dpda.*

Proof. Let M' be a super-nonsingular dpda in R_0 such that $L(M')=L(M)$, and let n_0 be a super-nonsingularity constant of M'. We assume, to the contrary, that M is not super-nonsingular. So, for any natural number n, there exists a strongly reachable pair of configurations $c_1=(p, w)$, $c_2=(p', ww')$ such that $\min(c_1)\geq\min(c_2)$ and $|w'|>n$. By Lemma 4.4 there exists a pair $\{d_1, d_2\}$ of configurations of M' satisfying (i)~(iii) of the lemma. By (i), $d_1\equiv c_1$ and $d_2\equiv c_2$, so we have $\min(d_1)=\min(c_1)\geq\min(c_2)=\min(d_2)$. Let $\beta_i\in L(d_i)$ such that $|\beta_i|=\min(d_i)$, $i=1, 2$. Consider the following computations from d_1 and d_2,

$$d_1 \downarrow(\delta_1)\, \bar{d}_1 \quad \text{and} \quad d_2 \to^{\delta_2} \bar{d}_2,$$

where δ_1 is a prefix of β_1 such that $|d_1|-|\bar{d}_1|=m+k_0$ and δ_2 is a prefix of β_2 such that $|\delta_2|=|\delta_1|$. (The computations are analogous to those shown in Fig. 2.) By the minimality of δ_1, $|\delta_1|\leq k_0(k_0+m)$ holds, so that the above computations ensure

$$|\bar{d}_2|\geq|d_2|-|\delta_2|\geq|d_2|-k_0(k_0+m), \qquad |\bar{d}_1|\leq|d_1|.$$

Since $|d_2|-|d_1|\geq l\log n$ by Lemma 4.4(ii), it follows that

$$|\bar{d}_2|-|\bar{d}_1|\geq l\log n-k_0(k+m) \;(=n_1).$$

So if we choose n so that $n_1>n_0$ holds, then $|\bar{d}_2|-|\bar{d}_1|>n_0$ and the pair $\{\bar{d}_1, \bar{d}_2\}$ is strongly reachable. Note that $\min(\bar{d}_1)\geq\min(\bar{d}_2)$ holds, because $\min(\bar{d}_i)=\min(d_i)-|\delta_i|$ for $i=1, 2$, $|\delta_1|=|\delta_2|$ and $\min(d_1)\geq\min(d_2)$. Hence, n_0 is not a super-nonsingularity constant of M', a contradiction. □

By Lemma 5.4, the class SN_0 is R_0-closed. Thus, containment (dpda in R_0, SN_0) reduces to the problem of deciding whether a dpda in R_0 is super-nonsingular. We will show that the latter problem is decidable. To show this we need some lemmas which are similar to [12, Lemmas 3.3 and 3.4]. Henceforth, we are dealing with a fixed dpda $M=(Q, \Gamma, \Sigma, \Delta, c_s, F)$ in R_0.

Definition 5.5 ([12]). For a configuration c, let $K(c)=\{(q, \varepsilon)\mid c\to^{\alpha}(q, \varepsilon)$ for some string $\alpha\}$.

Definition 5.6 ([12]). The nonnull segment v of a configuration $c=(q, wvu)$ is said to be loss-less for c if $K(q, u)=K(q, vu)$.

Note. If v is loss-less for $c=(q, wvu)$, then $K(q, u)=K(q, v^n u)$ for any $n\geq 0$.

Definition 5.7. A pair of reachable configurations $c_1=(q, wvu)$ and $c_2=(r, wvu)$ is said to be marked if segment v in Γ^+ is loss-less for both c_1, c_2 and there exists a reachable configuration $c_0=(p, wA)$ such that $c_0\uparrow(\gamma)(p, wvA)\rightarrow^{\gamma_1} c_1$ and $(p, wvA)\rightarrow^{\gamma_2} c_2$ for some input strings $\gamma, \gamma_1, \gamma_2$.

Lemma 5.8. *Let dpda M in R_0 be super-nonsingular and let $\{c_1=(q, wvu), c_2=(r, wvu)\}$ be a marked pair of reachable configurations. If $c_1\downarrow(\delta)(q_1, wv)\downarrow(\alpha)(q_1, w)$ for input strings δ, α and state q_1, then there exists a state $r_1\in K(r, u)$ and an input β such that $(r_1, v)\rightarrow^{\beta}(r_2, \varepsilon)$ and $|\beta|\leqslant|\alpha|$.*

Proof. Suppose, to the contrary, that for any state $r_1\in K(r, u)$ and any input β, if $(r_1, v)\rightarrow^{\beta}(r_2, \varepsilon)$, then $|\beta|>|\alpha|$. Since the pair $\{c_1, c_2\}$ is marked, for any $n>0$ there exists a reachable configuration $d_n=(p, wv^nA)$ such that $d_n\rightarrow^{\delta_1} c_{1n}=(q_1, wv^n)$ and $d_n\rightarrow^{\delta_2} c_{2n}=(r_1, wv^n)$ for some inputs δ_1, δ_2.

Note that the definition of α implies $c_{1n}\downarrow(\alpha^n)(q_1, w)$. Then we observe that if $c_{2n}\downarrow(\beta')(r', wv^{n-m})$, then $|\beta'|\geqslant|\alpha^m|+m$. Because, otherwise, since $K(r, u)=K(r, v^iu)$, $i\geqslant 0$, there must exist some state $r_1\in K(r, u)$ and some β such that $(r_1, v)\rightarrow^{\beta}(r_2, \varepsilon)$ and $|\beta|\leqslant|\alpha|$, but this is impossible from our assumption.

Let n_0 be a super-nonsingularity constant of dpda M. Let $c_{2n}\downarrow(\beta')\, c_{2k}=(r', wv^k)$ where $k=n-(n_0+1)$. Then, the pair $\{c_{2k}=(r', wv^k), c_{1n}=(q_1, wv^n)\}$ is strongly reachable and $|c_{1n}|-|c_{2k}|>n_0$ by $v\neq\varepsilon$ and $k=n-(n_0+1)$. So, if $\min(c_{2k})\geqslant\min(c_{1n})$, we have a contradiction. To show this, we observe that

$$\min(c_{1n})\leqslant k_0\cdot|w|+|\alpha^n|,$$

where k_0 is the constant $k_{0,M}$ of Definition 4.2. Since if $(r', wv^k)\downarrow(\beta'')(r'', w)$, then $|\beta''|\geqslant|\alpha^k|+k$, we have

$$\min(c_{2k})\geqslant|w|+|\alpha^k|+k.$$

So if we can choose n such that $|w|+|\alpha^k|+k\geqslant k_0|w|+|\alpha^n|$, that is, $k\geqslant(k_0-1)|w|+|\alpha^{n-k}|$, then $\min(c_{2k})\geqslant\min(c_{1n})$ holds. By $k=n-(n_0+1)$, the above inequality is equivalent to $n-(n_0+1)\geqslant(k_0-1)\cdot|w|+|\alpha^{n_0+1}|$. Hence, there exists a natural number n such that $\min(c_{2k})\geqslant\min(c_{1n})$. Thus, n_0 is not a super-nonsingularity constant of M, a contradiction. □

Lemma 5.9. *Let M be a super-nonsingular dpda in R_0. Let $l_1=k_0l_0^32^{2l_0}$ where $l_0=|M|$. Then, if a pair of configurations $c=(p, w)$ and $c'=(p', ww')$ is strongly reachable and $|w'|>l_1$, then $\min(c)<\min(c')$.*

Proof. The proof is the same as the proof of [12, Lemma 3.3]. In the proof of [12, Lemma 3.3], note that the assumption that M accepts a simple language is inessential and only [12, Lemma 3.2] corresponding to our Lemma 5.8 is necessary. So, using Lemma 5.8, the same proof is possible. □

The following theorem gives a characterization of super-nonsingular dpda's.

Theorem 5.10. *For a dpda M in R_0, the following three statements are equivalent:*

(i) *M is super-nonsingular.*

(ii) *M satisfies the following condition:*

(D) *For any pair of strongly reachable configurations $c_1 = (p, w)$ and $c_2 = (p', ww')$, if $\min(c_1) \geq \min(c_2)$, then $|w'| \leq l_1$ where l_1 is the constant of Lemma 5.9.*

(iii) *M satisfies the following condition:*

(E) *For any reachable configuration $c = (q, wA)$, A in Γ, if $c \downarrow(\alpha)\, c_1 = (p_1, w)$ and $c \downarrow(\beta)\, c_2 = (p_2, w)$, then $|\min(c_1) - \min(c_2)| \leq k_0(l_1 + 1)$.*

Proof. (i)$\Rightarrow$(ii). This is obvious by Lemma 5.9.

(ii)$\Rightarrow$(iii). Suppose, to the contrary, that M does not satisfy condition (E). Then, there exist configurations c_1, c_2 such that

$$c = (q, wA) \downarrow(\alpha)\, c_1 = (p_1, w), \qquad c = (q, wA) \downarrow(\beta)\, c_2 = (p_2, w),$$

where $|\min(c_1) - \min(c_2)| > k_0(l_1 + 1)$. Without loss of generality, let

$$\min(c_1) > \min(c_2) + k_0(l_1 + 1). \tag{$\bigstar$}$$

Let $\gamma_i \in L(c_i)$ such that $|\gamma_i| = \min(c_i)$ for $i = 1, 2$.

Let δ_1 be a prefix of γ_1 such that

$$c_1 \downarrow(\delta_1)\, c_1' \quad \text{and} \quad |c_1| - |c_1'| = l_1 + 1.$$

Then, $|\delta_1| \leq k_0(l_1 + 1)$ holds, since δ_1 is minimal for c_1. Hence,

$$\begin{aligned}\min(c_1') &= \min(c_1) - |\delta_1| \geq \min(c_1) - k_0(l_1 + 1)\\ &> \min(c_2) \quad \text{(by inequality } (\bigstar)).\end{aligned}$$

But, $|c_2| - |c_1'| > l_1$ holds by $|c_2| = |c_1|$ and $|c_1| - |c_1'| = l_1 + 1$. Thus, $\{c_2, c_1'\}$ is a pair of strongly reachable configurations such that $\min(c_1') > \min(c_2)$ and $|c_2| - |c_1'| > l_1$, that is, M does not satisfy condition (D), a contradiction.

(iii)$\Rightarrow$(i). Let $\{c_1 = (p, w), c_2 = (p', ww')\}$ be a pair of strongly reachable configurations such that

$$\min(c_1) \geq \min(c_2). \tag{$\bigstar_1$}$$

To show that $|w'|$ is bounded by a fixed constant, let $\gamma_2 \in L(c_2)$ such that $|\gamma_2| = \min(c_2)$, and consider the following computation:

$$c_2 = (p', ww') \downarrow(\delta_2)\, c_2' = (q, w)$$

for a prefix δ_2 of γ_2. Note that $\min(c_2) = \min(c_2') + |\delta_2|$ and $|\delta_2| \geq |w'|$, since dpda M is real-time. So by inequality ($\bigstar_1$) we have

$$\min(c_1) \geq \min(c_2') + |\delta_2| \geq \min(c_2') + |w'|. \tag{$\bigstar_2$}$$

Since dpda M satisfies condition (E), $|\min(c_1)-\min(c_2')|\leq k_0(l_1+1)$ holds for $c_1=(p,w)$ and $c_2'=(q,w)$, so that, by inequality ($\bigstar_2$), $|w'|\leq k_0(l_1+1)$ holds. Hence, $k_0(l_1+1)$ is a super-nonsingularity constant of M. □

We can show that it is decidable whether a dpda in R_0 satisfies condition (E) of Theorem 5.10(iii). To show this, from a given dpda M in R_0 we construct a dpda M' such that $L(M')=\emptyset$ iff M satisfies condition (E). Thus, whether condition (E) holds reduces to the emptiness problem for dpda's, which is known to be decidable.

The computations of M' simulate M and simultaneously check whether M satisfies condition (E). That is, if M makes a computation $c_s \rightarrow^{\alpha} c=(q, A_1A_2\ldots A_n)$ for input string α, A_i in Γ, $1\leq i\leq n$, then M' makes $c_s' \rightarrow^{\alpha} c'=(q, A_1'A_2'\ldots A_n')$. Here, c_s' is the initial configuration of M', and $A_i'=[A_i, S_i]$, $1\leq i\leq n$, is a stack symbol of M' where S_i saves information necessary for check. More precisely, S_i keeps track of differences between $\min(c_{i_1}),\ldots,\min(c_{i_k})$ of configurations $c_{i_1},\ldots,c_{i_k}$ where $c_{i_j}=(r_j, A_1\ldots A_{i-1})$ for some state r_j, $1\leq j\leq k$, and $c_{i_1},\ldots,c_{i_k}$ are all the configurations reachable from the configuration at the last time that ith symbol A_i of configuration c was stacked. If by examining S_i, dpda M' discovers that $\min(c_{i_l})-\min(c_{i_{l'}})>k_0(l_1+1)$ for some configurations c_{i_l} and $c_{i_{l'}}$, then M' enters an accepting configuration. Here, computations from accepting configurations of M' are undefined. Thus, $L(M')=\emptyset$ iff M satisfies condition (E).

The remaining problem is how to calculate S_i with information for check. Formally, S_i is represented by $\{(r_1,m_1),\ldots,(r_k,m_k)\}\subseteq Q\times\mathbb{N}$ where there exists an l ($1\leqq l\leqq k$) such that $m_l=0$, and for each j ($1\leq j\leq k$), m_j is an offset from the base $\min(c_{i_l})$, that is, $m_j=\min(c_{i_j})-\min(c_{i_l})$. Here, $c_{i_j}=(r_j, A_1\ldots A_{i-1})$, $1\leq j\leq k$, and $m_l=0$ implies that $\min(c_{i_l})$ is minimal.

We are now ready to explain how to calculate S_i. Let $c_s'\rightarrow^{\alpha} c'=(q, A_1'\ldots A_n')$ where $A_i'=[A_i,S_i]$, $1\leq i\leq n$, and $c_s\rightarrow^{\alpha} c=(q, A_1\ldots A_n)$. For an input a in Σ, let $(q, A_n)\rightarrow^{a}(p, B_1\ldots B_m)$ be a computation of M. Assuming that $S_1,\ldots,S_n$ were correctly calculated, the new S_i's are calculated as follows. If $m=0$, then the corresponding computation of M' is $c'\rightarrow^{a}(p, A_1'\ldots A_{n-2}'A_{n-1}'')$ where $A_{n-1}''=(A_{n-1}, S_{n-1}')$ and S_{n-1}' is a subset of S_{n-1} where $(r,m)\in S_{n-1}'$ iff $(r,m)\in S_{n-1}$ and configuration $(r, A_1\ldots A_{n-2})$ is reachable from $(p, A_1\ldots A_{n-2}A_{n-1})$. Note that if there exists no state r such that $(r,0)\in S_{n-1}'$, then each pair $(q,m)\in S_{n-1}'$ is replaced by $(q, m-m')$ where $m'=\mathrm{Min}\{m_j\,|\,(w_j,m_j)\in S_{n-1}'\}$. (We assume that such a modification is made whenever needed, hereafter.)

If $m>0$, then the corresponding computation of M' is $c'\rightarrow^{a}(p, A_1'\ldots A_{n-1}'B_1'\ldots B_m')$, where $B_i'=[B_i, S_i'']$ for $1\leq i\leq m$, is calculated as follows: S_1'' is a subset of S_n and the calculation is similar to that in the case where $m=0$, and therefore omitted. For S_j'', $2\leq j\leq m$, it is calculated from S_{j-1}'' as follows: $(r,l)\in S_j''$ iff $r\in K(p, B_j\ldots B_m)$ and

$$l=\mathrm{Min}\{|\beta|+l'\,|\,(r, B_{j-1})\rightarrow^{\beta}(r',\varepsilon)\text{ and }(r',l')\in S_{j-1}''\}.$$

It is obvious that S_j'' can be calculated from S_{j-1}''. Further, $(r,l)\in S_j''$ implies that

$\min(r, A_1 \ldots A_{n-1}B_1 \ldots B_{j-1}) = l + \text{base}$ where $\text{base} = \min(r'', A_l \ldots A_{n-1}B_1 \ldots B_{j-2})$ and $(r'', 0) \in S''_{j-1}$. Thus, S''_j is a correct representation of differences between configurations $(r, A_1 \ldots A_{n-1}B_1 \ldots B_{j-1})$, $r \in K(p, B_j \ldots b_n)$. Since it is obvious that M' is a dpda, by the above arguments we have the following theorem.

Theorem 5.11. *It is decidable, for a given dpda M in R_0, whether M is super-nonsingular.*

By Lemma 5.4 and Theorem 5.11 we have the following lemma.

Lemma 5.12. *It is decidable, for a given dpda M in R_0, whether M accepts a super-nonsingular language.*

Since containment (dpda, SN_0) reduces to containment (dpda in R_0, SN_0) by $\mathscr{L}(SN_0) \subseteq \mathscr{L}(R_0)$ and [11, Theorem 2], we have the main theorem of this section.

Theorem 5.13. *Containment (dpda, SN_0) is decidable.*

In Definition 5.1 of super-nonsingular dpda's, $\min(c_1) \geq \min(c_2)$ in property (C) can be replaced by $\min(c_1) = \min(c_2)$. It is straightforward that the new definition is equivalent to the old one. Thus, it is decidable, for a dpda M in R_0, whether for any pair of strongly reachable configurations c and c' of M, the difference between $|c|$ and $|c'|$ is bounded by a fixed constant under $\min(c) = \min(c')$. We can compare this result with the undecidability result of Theorem 3.11 of Section 3. We observe that the above problem under $c \equiv c'$ instead of $\min(c) = \min(c')$ is undecidable.

The definition of super-nonsingular dpda's may be artificial, but the class includes important subclasses such as, simple dpda's and LL(k) acceptors by Lemma 5.3. So containment (dpda, LL(k)), which remains open, reduces to containment (dpda in $SN_0 \cap R_0$, LL(k)).

Corollary 5.14. *Containment (dpda, LL(k)) reduces to containment (dpda in $SN_0 \cap R_0$, LL(k)).*

Proof. The proof follows from [11, Theorem 2] and our Lemmas 5.12 and 5.4. □

Thus, the above result concerning super-nonsingular dpda's may be useful for resolving containment (dpda, LL(k)).

Acknowledgment

The author would like to thank Professor Namio Honda, Toyohashi University of Technology, Professor Teruo Fukumura and Professor Yasuyoshi Inagaki, Nagoya University for their encouragement and advice. He is also grateful to the referees for their critical reading of the earlier versions of this paper, and for giving many subsequent helpful suggestions.

References

[1] B. Courcelle, On jump-deterministic pushdown automata, *Math. Systems Theory* **11** (1977) 87–109.
[2] E.P. Friedman, A note on nonsingular deterministic pushdown automata, *Theoret. Comput. Sci.* **7** (1978) 333–339.
[3] E.P. Friedman and S.A. Greibach, On equivalence and subclass containment problems for deterministic context-free languages, *Inform. Process. Lett.* **7** (1978) 287–291.
[4] E.P. Friedman and S.A. Greibach, Superdeterministic DPDAs: The method of accepting does affect decision problems, *J. Comput. System Sci.* **19** (1979) 79–117.
[5] S.A. Greibach, Linearity is polynomially decidable for realtime pushdown store automata, *Inform. Contr.* **42** (1979) 27–37.
[6] M.A. Harrison and I.M. Havel, Real-time strict deterministic languages, *SIAM J. Comput.* **1** (1972) 333–349.
[7] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation* (Addison-Wesley, Reading, MA, 1979).
[8] A.J. Korenjak and J.E. Hopcroft, Simple deterministic languages, *IEEE 7th Symp. on Switching and Automata Theory* (1966) 36–46.
[9] M. Oyamaguchi, Y. Inagaki and N. Honda, The equivalence problem for real-time strict deterministic languages, *Inform. Contr.* **45** (1980) 90–115.
[10] M. Oyamaguchi, Y. Inagaki and N. Honda, On the equivalence problem for two dpda's, one of which is real-time, *Information Processing 80* (IFIP–North-Holland, Amsterdam, 1980) 53–57.
[11] M. Oyamaguchi, Y. Inagaki and N. Honda, A real-time strictness test for deterministic pushdown automata, *Inform. Contr.* **47** (1980) 1–9.
[12] M. Oyamaguchi, Y. Inagaki and N. Honda, A simplicity test for deterministic pushdown automata, *Inform. Contr.* **47** (1980) 10–36.
[13] M. Oyamaguchi, Y. Inagaki and N. Honda, The equivalence problem for two dpda's, one of which is a finite-turn or one-counter machine, *J. Comput. System Sci.* **23** (1981) 366–382.
[14] D.Z. Rosenkrantz and R.E. Stearns, Properties of deterministic topdown grammars, *Inform. Contr.* **17** (1970) 226–256.
[15] R.E. Stearns, A regularity test for pushdown machines, *Inform. Contr.* **11** (1967) 323–340.
[16] K. Taniguchi and T. Kasami, A result on the equivalence problem for deterministic pushdown automata, *J. Comput. System Sci.* **13** (1976) 38–50.
[17] E. Tomita, A branching algorithm for deciding equivalence of deterministic pushdown automata (in D_0; R_0), *Trans. IECE Japan* **J61-D** (1978) 759–766.
[18] L.G. Valiant, Decision procedures for families of deterministic pushdown automata, Ph.D. Thesis, University of Warwick, 1973.
[19] L.G. Valiant, The equivalence problem for deterministic finite-turn pushdown automata, *Inform. Contr.* **25** (1974) 123–133.
[20] L.G. Valiant, Regularity and related problems for deterministic pushdown automata, *J. Assoc. Comput. Mach.* **22** (1975) 1–10.
[21] B. Courcelle, The simultaneous accessibility of two configurations of two equivalent dpda's, *Inform. Process. Lett.* **12** (1981) 111–114.
[22] E. Tomita, A direct branching algorithm for checking equivalence of some classes of deterministic pushdown automata, *Inform. Contr.* **52** (1982).